Requirements for Scientific Data Management

Suren Byna

Scientific Data Management Group
Computational Research Division
Lawrence Berkeley Lab

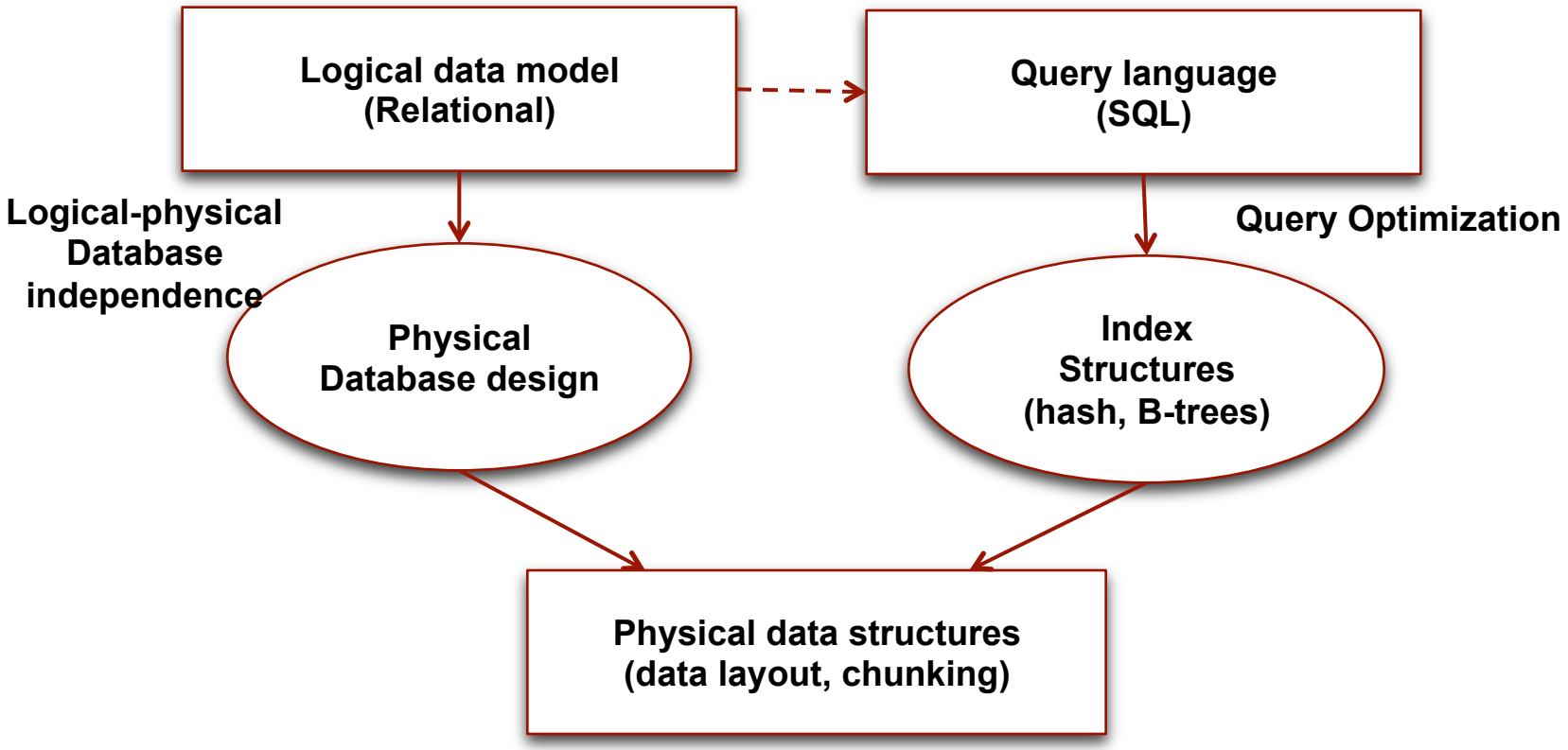
NERSC ASCR Requirements for 2017
January 15, 2014
LBNL

Projects

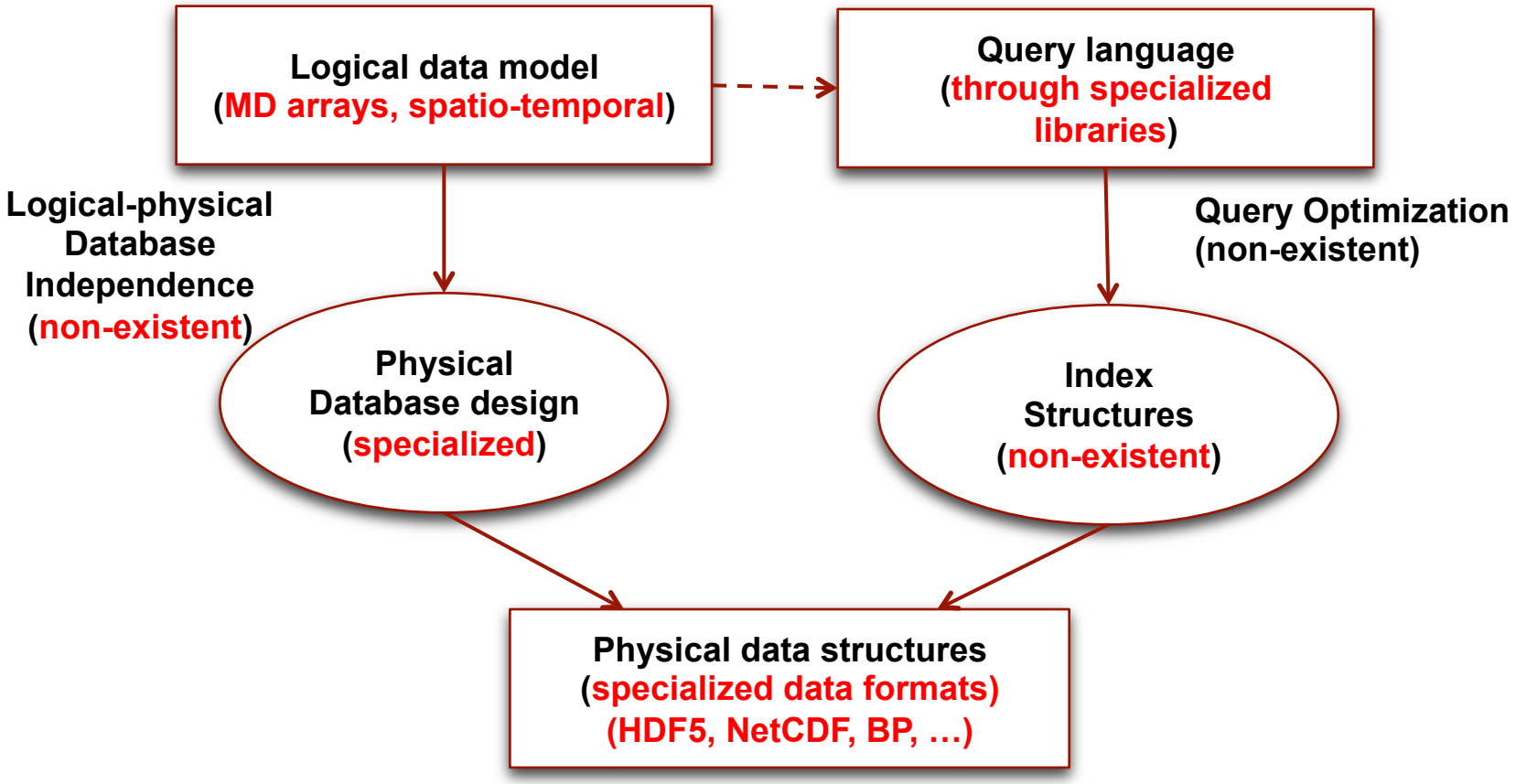
- m1248 repo
  - Arie Shoshani, Suren Byna, Alex Sim, John Wu
- Searching scientific data
  - FastBit and FastQuery

- Scientific Data Services (SDS) framework
  - Transparent data reorganization for better data access
  - Redirection of data read calls for reorganized datasets
- Support for *in situ* and in transit analysis
  - International Collaboration Framework for Extreme Scale Experiments (ICEE)
- Improved parallel I/O performance
  - ExaHDF5

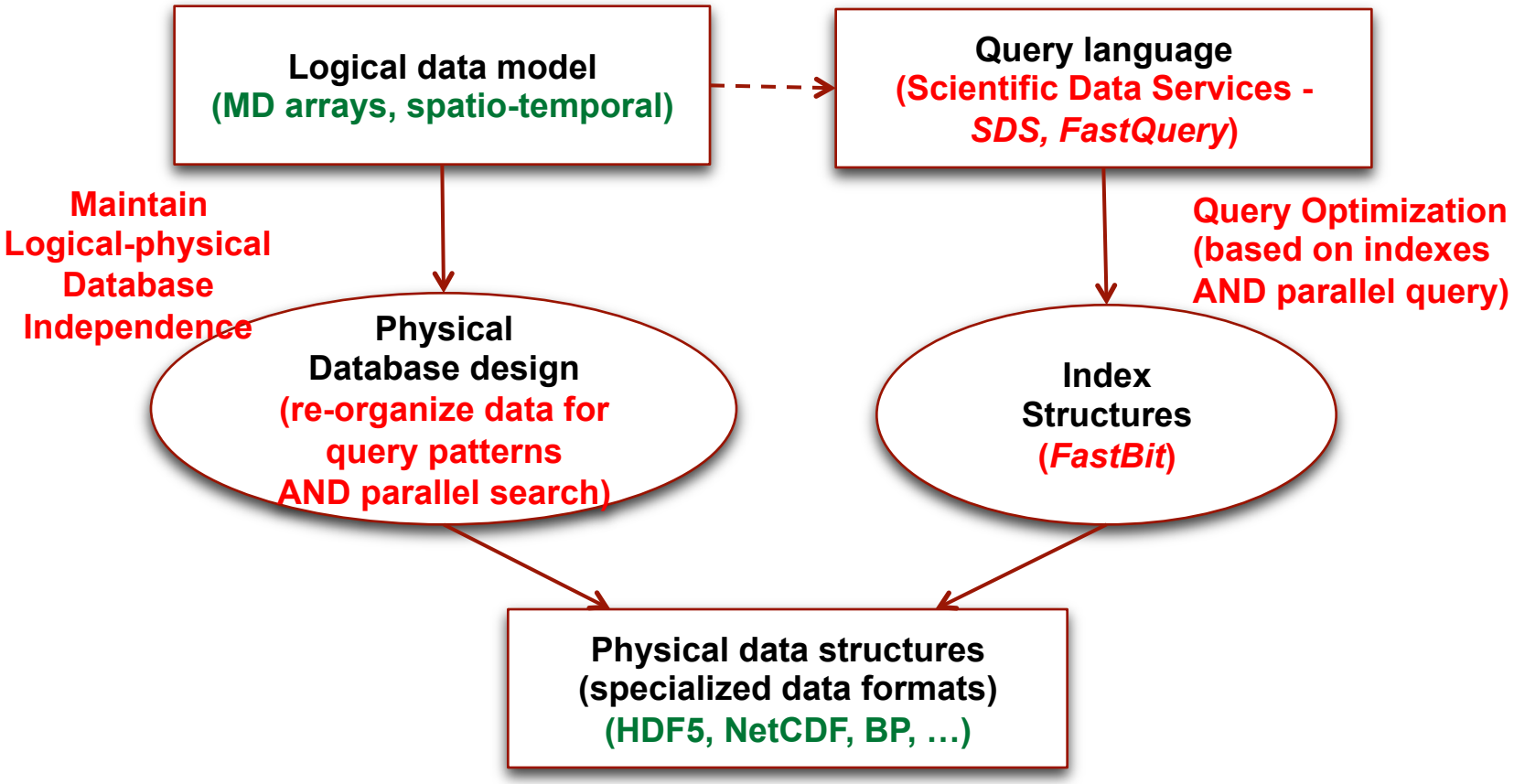
Traditional Data Management



Scientific Data Management – current state

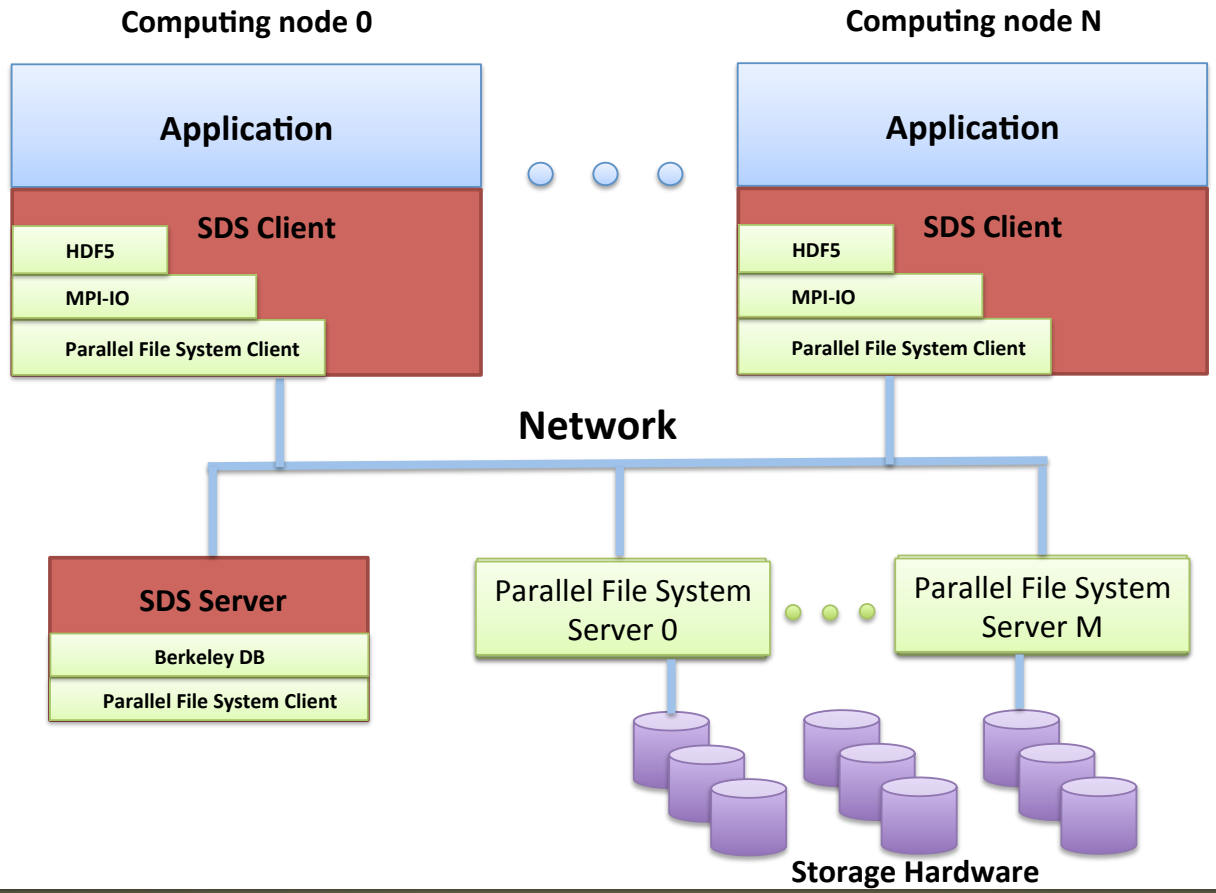


Scientific Data Management – filling the gap



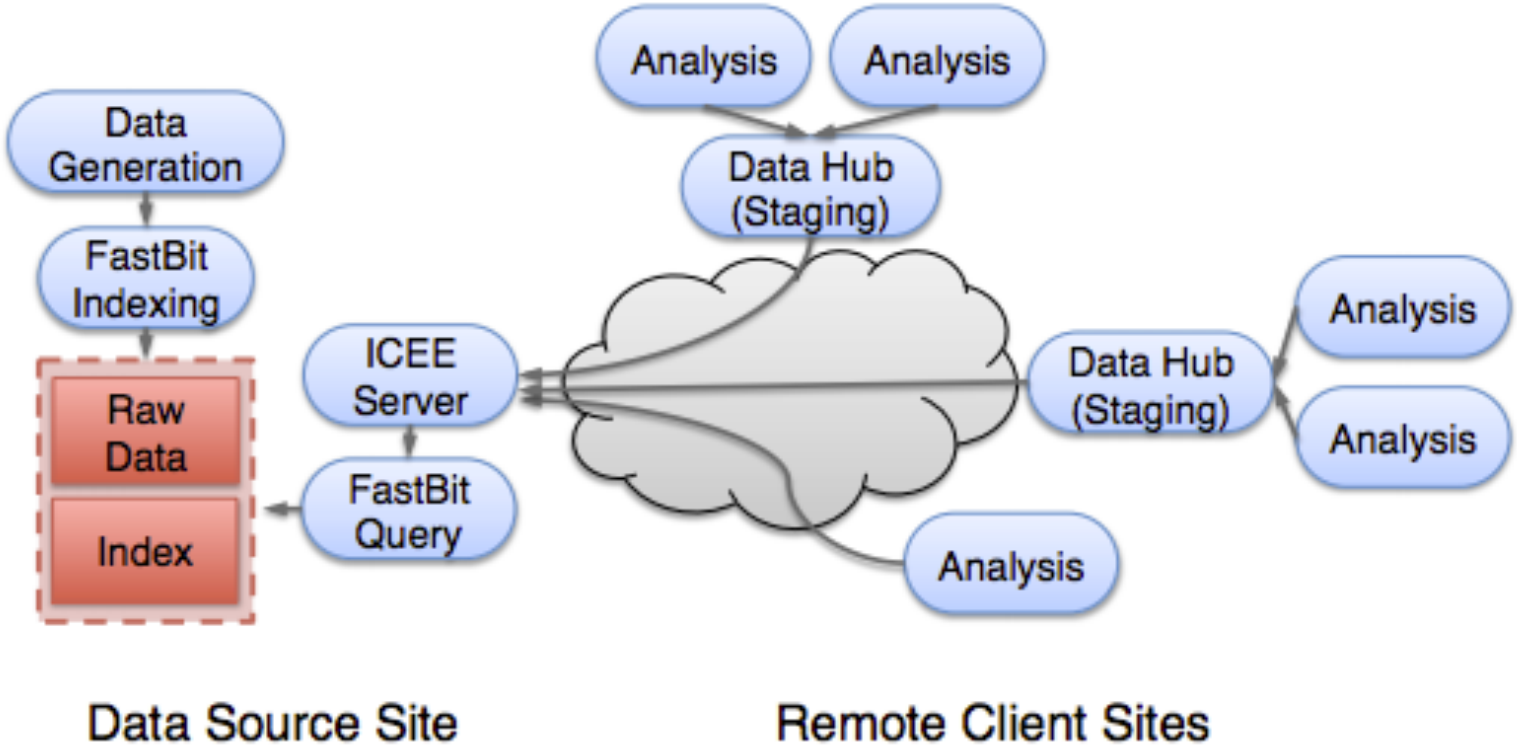
Scientific Data Services (SDS) Framework

- A framework to bring the merits of database management systems and parallel file systems together



In situ and In transit analysis

- Support for analyzing data while the data is being produced and is in transit before moving the data to destination



2. Computational Strategies

- Computational strategies for SDS
  - Parallel sorting → Trillions of particles, ~20,000 cores
  - Index generation (bitmap indexes, B-Tree indexes) → Trillions of particles/ elements, 10s of thousands of cores
  - Query optimization (hash-joins) → ~ billions of elements
- Codes
  - VPIC → Trillions of particles, ~150,000 cores
  - Mass Spectrometry Imaging analysis tasks
  - Palomar Transient Factory (PTF) → Hundreds of millions of records
  - Catalog matching → Hundreds of millions of records
  - I/O kernels → ~50TB, ~100,000 cores
- By 2017:
  - The sizes of the datasets will be in 100s of TB to 10s of PB
  - Data staging for supporting in situ and in transit analysis
  - Dedicated servers for persistent SDS services

3. Current HPC Usage (1 of 2)

- Machines currently using (NERSC or elsewhere)
  - Hopper, Edison
- Hours used in 2012-2013 (list different facilities)
  - ~3 million hours
- Typical parallel concurrency and run time, number of runs per year
  - Bitmap indexing: up to 20,000 cores
  - Parallel sorting: up to 10,000 cores
- Data read/written per run
  - Bitmap indexing: 0.3X to 3X the original data size
  - E.g. Index of one trillion particle data: ~150 TB for 12 time steps

3. Current HPC Usage (2 of 2)

- Memory used per (node | core | globally)
  - For VPIC simulation: 90% of the total memory on Hopper
  - Index generation: 75% of memory on each node

- Data resources used (/scratch,HPSS, NERSC Global File System, etc.) and amount of data stored
  - ~100 TB particle data on HPSS
  - ~100 TB particle indexes data on /project
  - ~ 50 TB /scratch space on Hopper for testing SDS

4. HPC Requirements for 2017

(Key point is to directly link NERSC requirements to science goals)

- Compute hours needed (in units of Hopper hours)
  - Project 5X increase in our usage
- Data needs
  - Similar to Prabhat's projection

	Compute Hours	Target Concurrency	Data read/ written per run	Memory per node	Required software	Resources used	Data Stored
Current 2014	4M	10K-150K	100GB-30TB	100%	HDF5, NetCDF, MPI, MPI-IO, pthreads, OpenMP, ScalaPACK, BLAS	/scratch /project	250-500 TB
Estimated 2017	30M	10K-10M	100GB-1PB	100%	HDF5, NetCDF, MPI, MPI-IO, MPI+X?? ScalaPACK, BLAS	/scratch /project Burst Buffers	1-5 PB

5. Strategies for New Architectures (1 of 2)

Does your software have CUDA/OpenCL directives; if yes, are they used, and if not, are there plans for this?

- Depending on analysis codes, SDS framework and ICEE plan to use heterogeneous processing

Does your software have OpenMP directives now; if yes, are they used, and if not, are there plans for this?

- FastQuery uses MPI + Pthreads
- VPIC code uses MPI+OpenMP

Is porting to, and optimizing for, the Intel MIC architecture underway or planned?

- Not planned, but considering their usage for data management and computing operators

5. Strategies for New Architectures (2 of 2)

- What role should NERSC/DOE/ASCR/ play in the transition to these architectures?
  - Explore new architectures for co-locating simulation and analysis
  - Explore new architectures for dedicated nodes for smart management of data movement
  - Fund efforts for energy efficient data management

- Other needs or considerations or comments on transition to manycore:
  - Performance and power consumption monitoring at CPU and system levels is key for identifying bottlenecks
  - ***(Not related to multicore)*** Performance monitoring at file system level is needed for improving parallel I/O
  - ***(Not related to multicore)*** Power consumption monitoring at storage system level is needed for improving energy efficiency of data movement

5. Special I/O Needs

- Does your code use checkpoint/restart capability now?
  - Simulations such as VPIC and AMR codes need fast checkpoint/restart capabilities

- Do you foresee that a burst buffer architecture would provide significant benefit to you or users of your code?
  - Burst buffers will be useful for *in situ* and in transit analysis when memory is not sufficient
  - SDS framework can use burst buffer or storage in staging area for prefetching and reorganizing data

6. Summary

- What new science results might be afforded by improvements in NERSC computing hardware, software and services?
  - Accelerate analysis codes that could lead to faster data to discovery
- What "expanded HPC resources" are important for your project?
  - Dedicated nodes for offering Scientific Data Services
  - NVM/NVRAM for analysis data that does not fit in memory and for prefetching data for future use
  - Faster file systems

Backup Slides

Scientific Data Services (SDS)

- A service to bring the merits of database management systems and parallel file systems together
- Programming interface for accessing arrays and for executing queries
- An optimization interface between data format libraries (HDF5, NetCDF, and ADIOS-BP) and file system optimizations
- Research activities:
  - Optimizations for accessing data in post-process phase via data reorganization: replicate data in different organizations for accesses
  - In-memory data processing and querying
    - ✓ Query optimization
    - ✓ In memory Indexing
  - Runtime support for deep memory hierarchies

Scientific Data Services (SDS) Benefit

- A data model familiar to domain scientists
- Existing file formats and analysis tools can co-exist with the new system
- Optimization of data access based on queries on data model
- Dynamic reorganization of data based on access patterns
- Large energy savings by accessing only data needed from disk
- Reducing data movement in memory using in-memory indexing, thus reducing energy usage
- Easier to integrate with analysis and visualization tools – integration can now be done at the data model level